Experimental Physics and Industrial Control System (EPICS) Overview

Bob Dalesio, Feb. 25,10,2015

Outline

- Introduction: What is EPICS
  - The Collaboration
  - Architecture
  - Tool-Kit
- IOC Core
  - Channel Access
  - Process Database
- Conclusions

What is EPICS?

- A collaboration of the controls groups of many research organizations that use the EPICS tool-kit.
- A distributed architecture that supports a wide range of solutions from small test stands to large integrated facilities.
- A set of tools that reduces software application and maintenance costs by providing:

  Configuration tools in place of programming

  A large installed base of tested software

  A modular design that supports incremental upgrades

  Well defined interfaces for extensions at every level

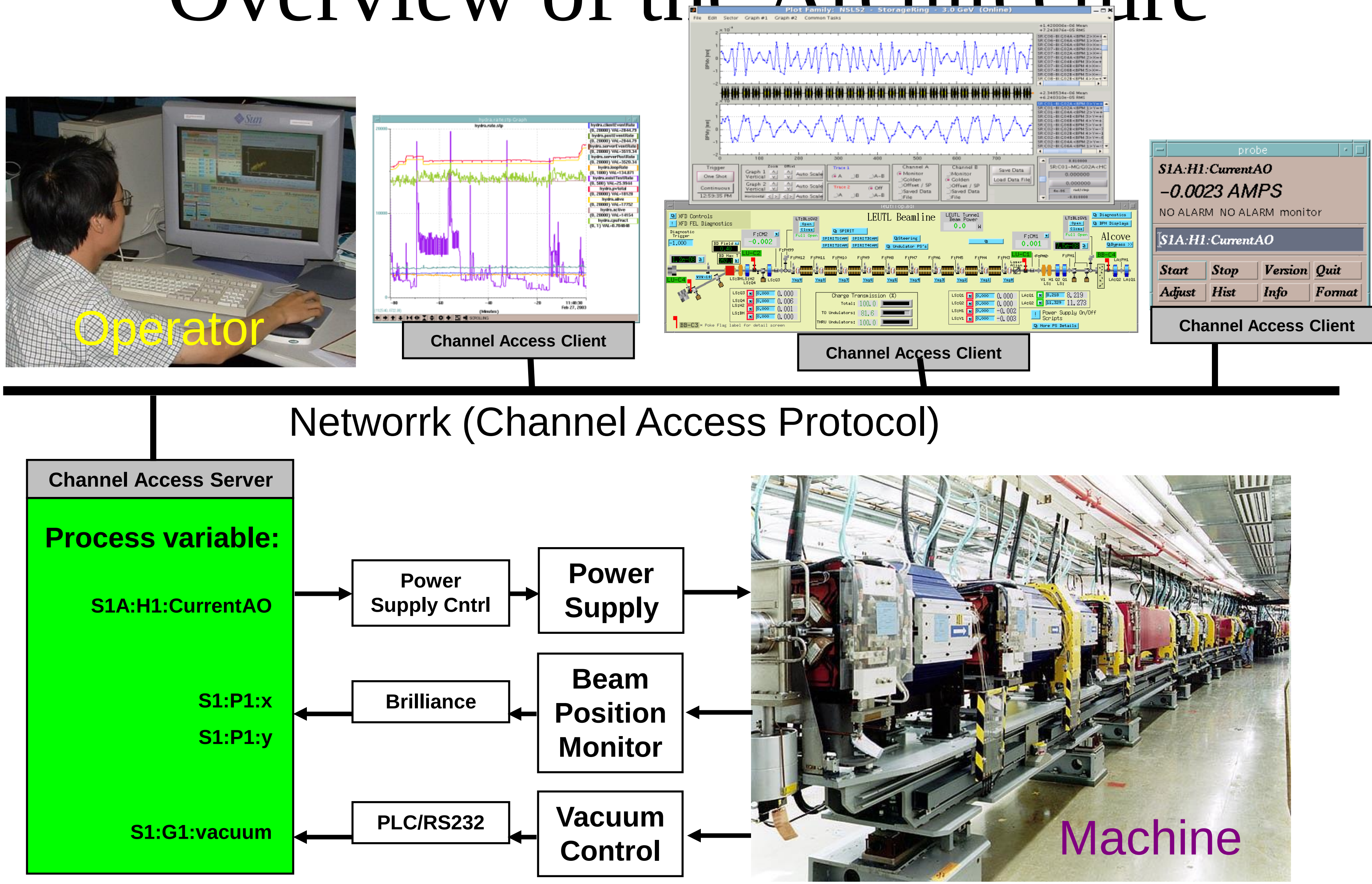
What is EPICS? The Collaboration

- Over 150 independent projects in North America, Europe, Africa, Australia, South America, and Asia
- Applications in particle physics, astronomy, and industrial control
- Distribute software over the network
- Independent development, co-development and incremental development of code done by members
- Problem reporting and resolution via e-mail exploders
- Documentation available on WWW sites
- Large collaboration meetings to report new work, discuss future directions, explore new applications, and explore new requirements for existing codes
- Small design groups from multiple labs meet to discuss design issues on significant codes: Channel Access, Archiving and MMI

What is EPICS? Distributed Architecture

- EPICS V3 is physically a flat architecture of front-end controllers and operator workstations that communicate via TCP/IP and UDP
    - System scales through the addition of new computers
    - Physical hierarchy is made through bridges, routers, or a gateway
    - Network bandwidth is the primary limiting factor
- EPICS software architecture is client/server based - with independent data stores providing read/write access directly between any two points
    - Local name services mean automatic integration of new components
    - Point-to-point communication supports automation

Overview of the Architecture



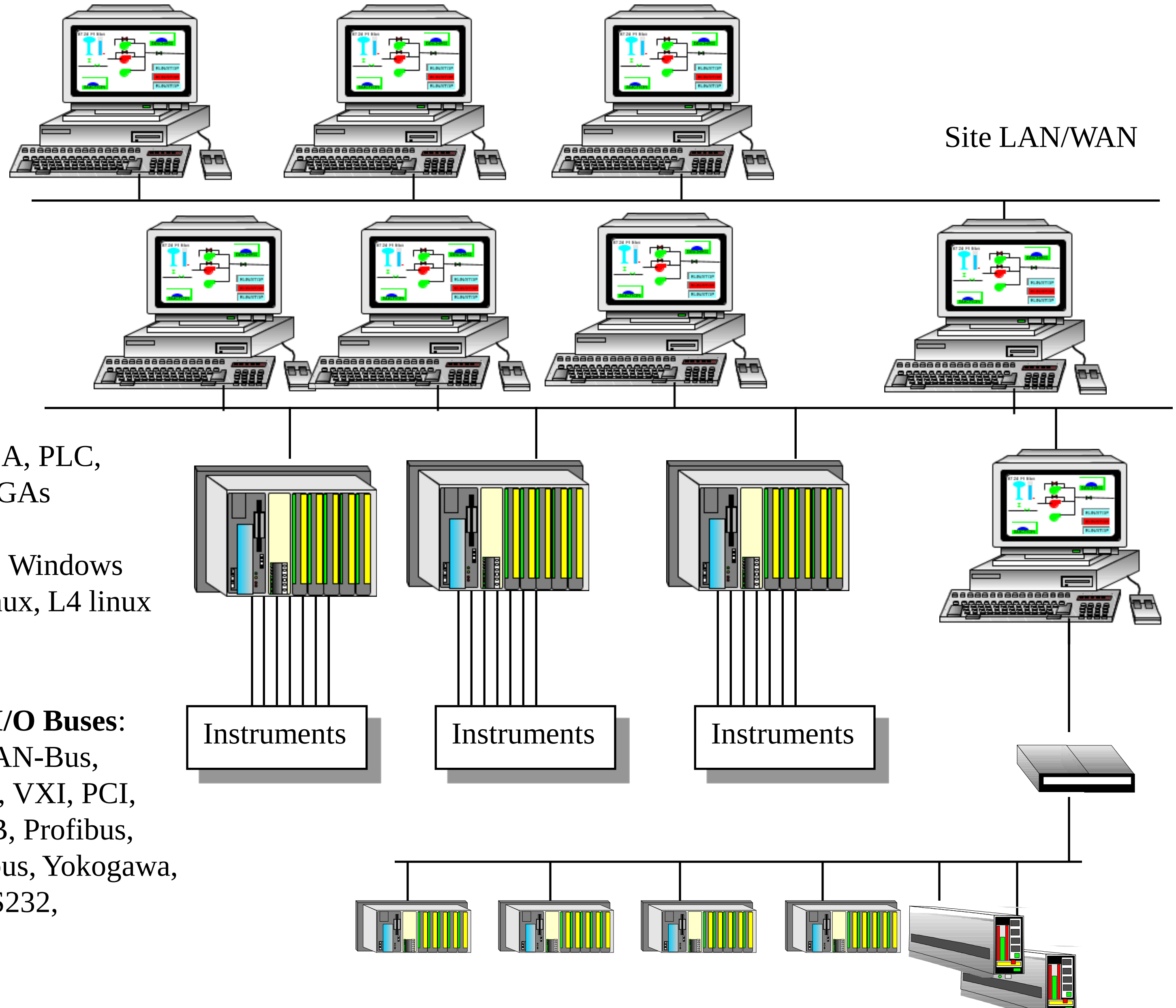
EPICS Supports a Standard Control System Architecture

OS:
Unix
Windows

I/O Controllers:
VME,VXI
PCI, PCIe, uTCA, PLC,
Softcores on FPGAs

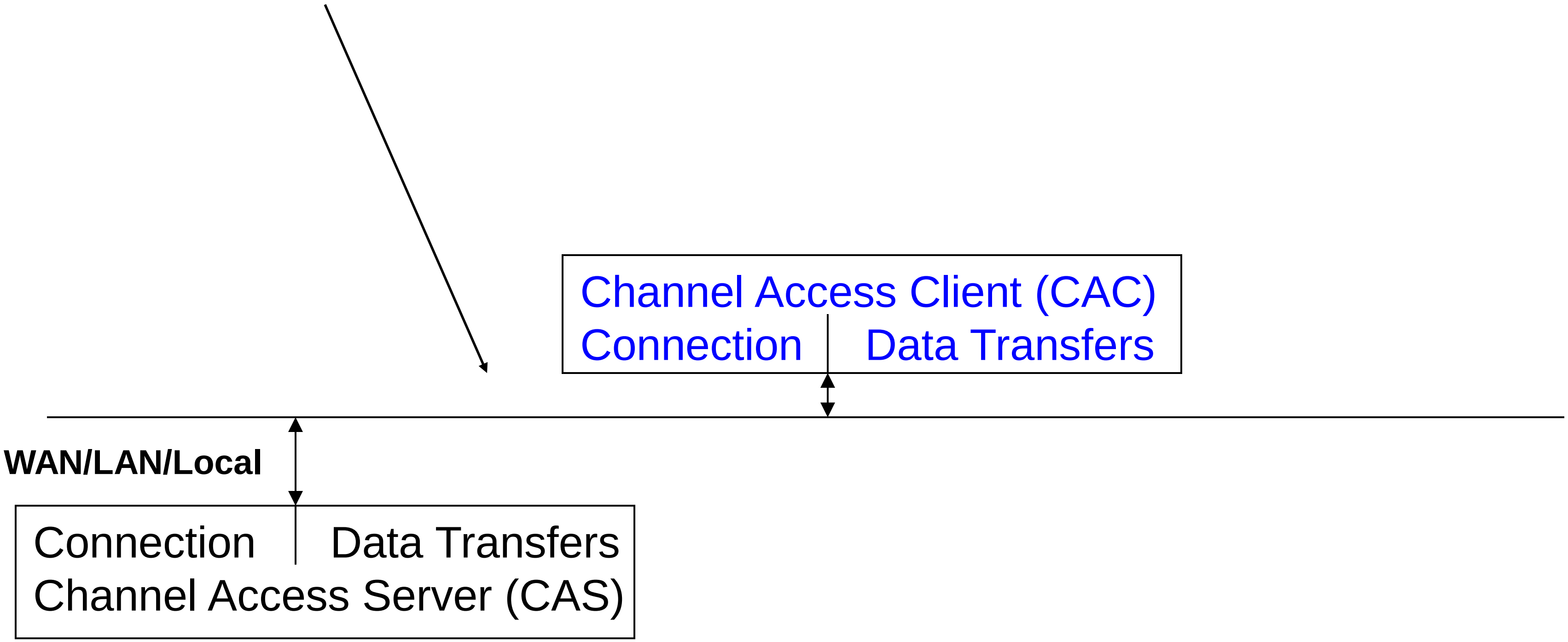
OS:
vxWorks, Unix, Windows
RTEMS, RTLinux, L4 linux

Remote and Local I/O Buses:
Control Net, PCI, CAN-Bus,
Industry Pack, VME, VXI, PCI,
ISA, CAMAC, GPIB, Profibus,
Bitbus, Serial, Modbus, Yokogawa,
G-3, Ethernet/IP, RS232,
Many more….

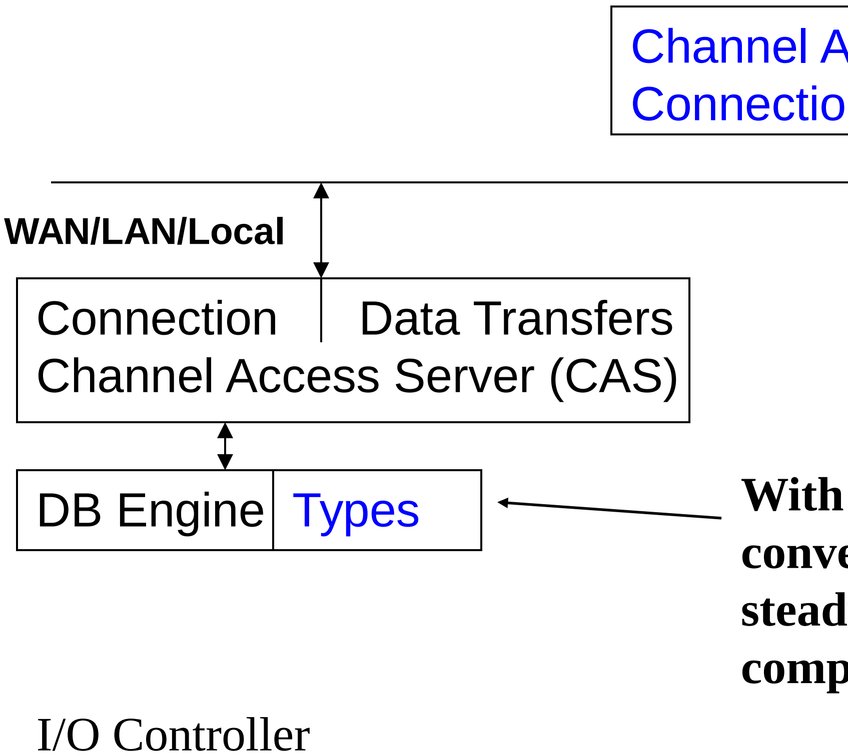


EPICS Architecture

Is based on the channel access protocol



EPICS Architecture



With a process database engine that describes conversions to engineering units, alarm limits, steady state logic for PID control, interlocks, and complex data conversions or filtering

EPICS Architecture

Channel Access Client (CAC)

Connection | Data Transfers

WAN/LAN/Local

Connection | Data Transfers

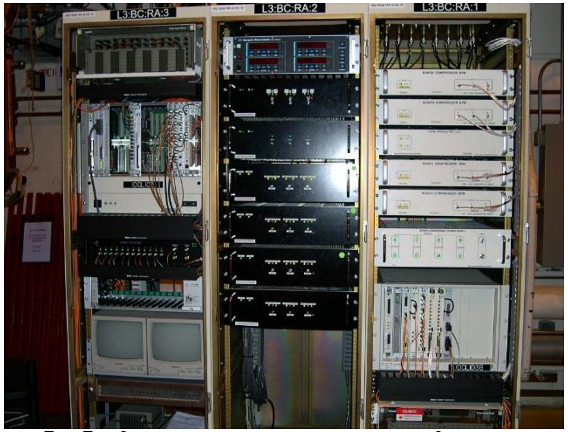
Channel Access Server (CAS)

DB Engine | Types

Device Support

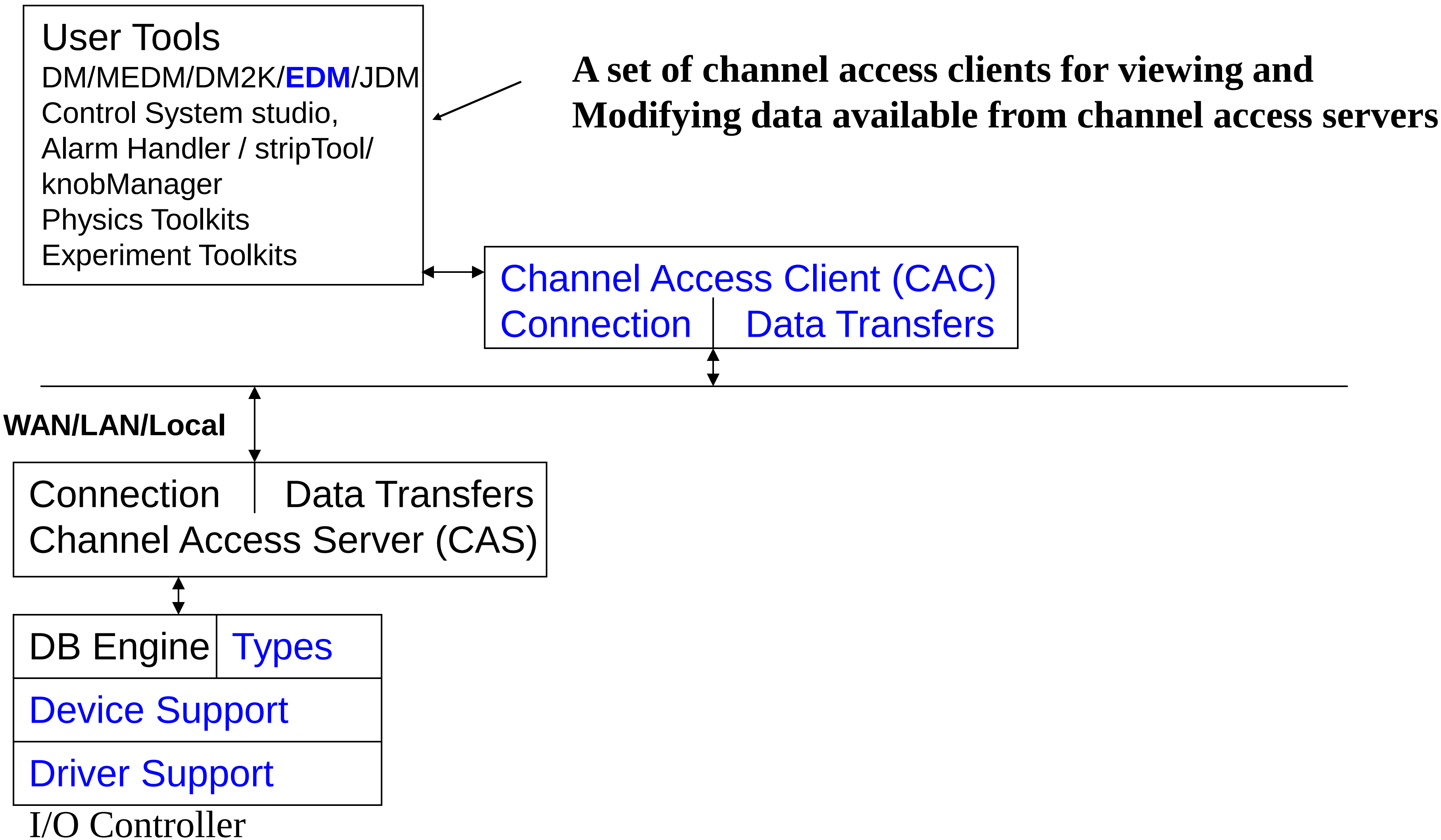
Driver Support

I/O Controller

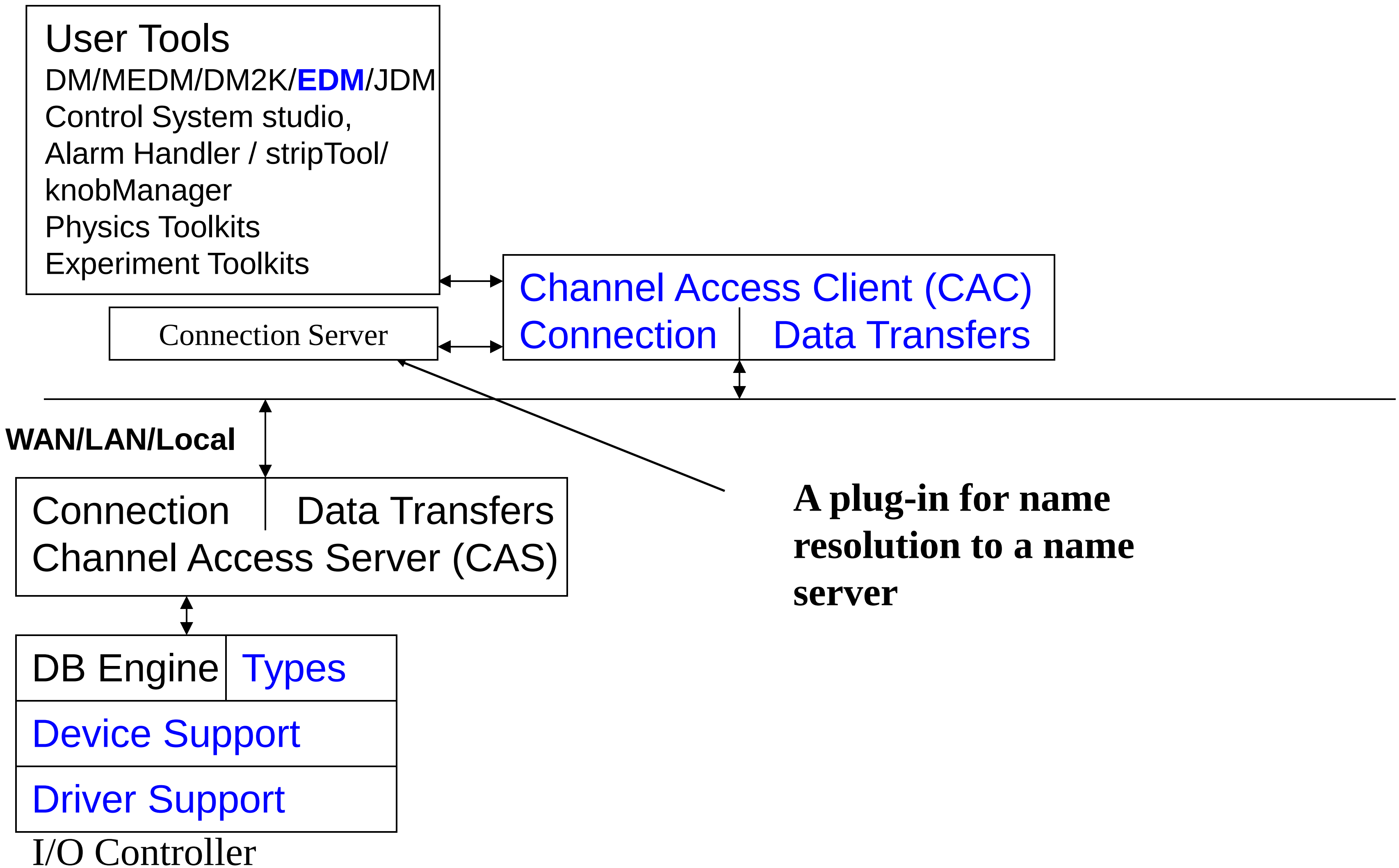


Device and driver support to integrate protocols required to communicate with a large variety of instrumentation and intelligent devices.

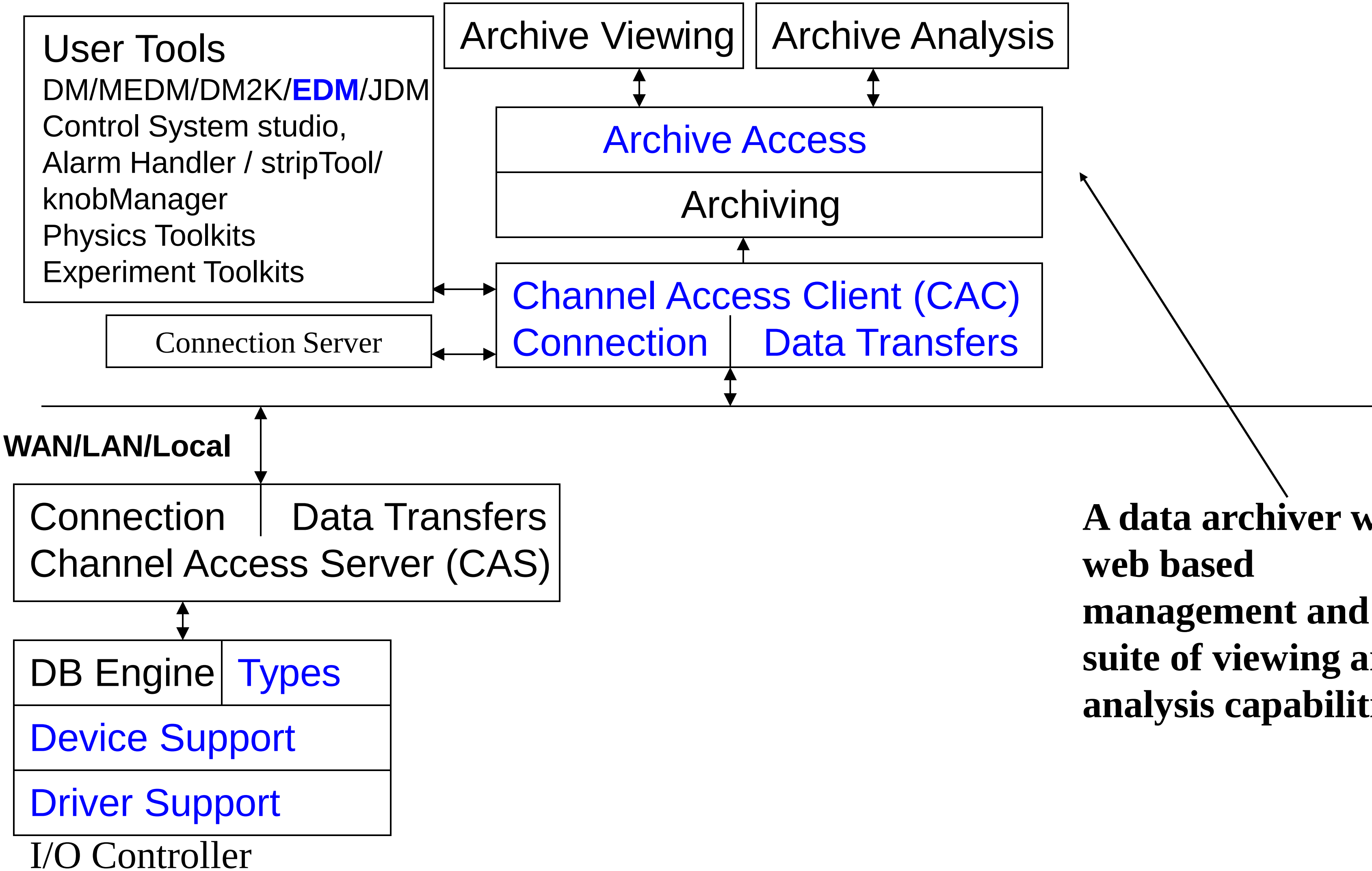
EPICS Architecture



EPICS Architecture

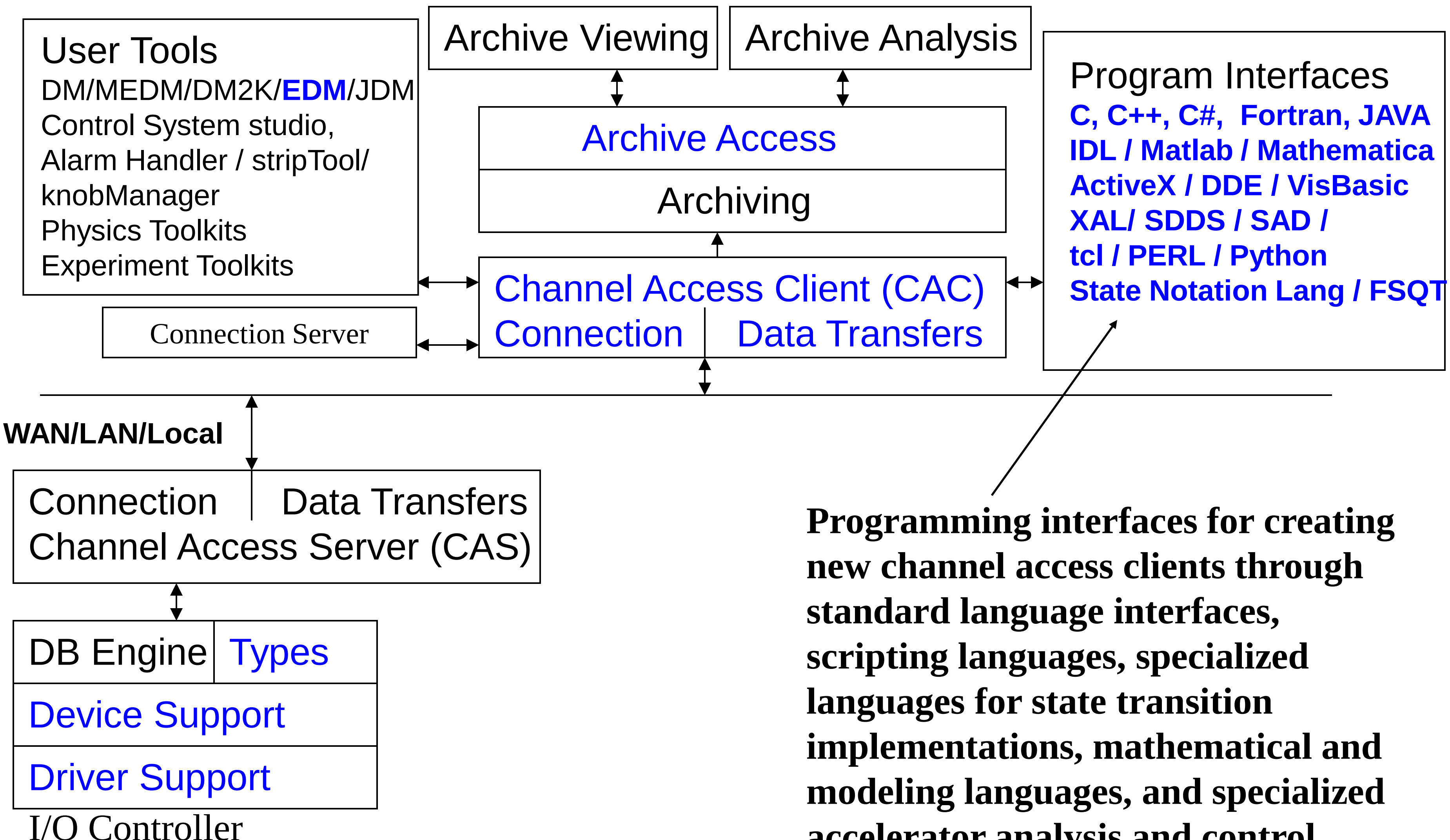


EPICS Architecture



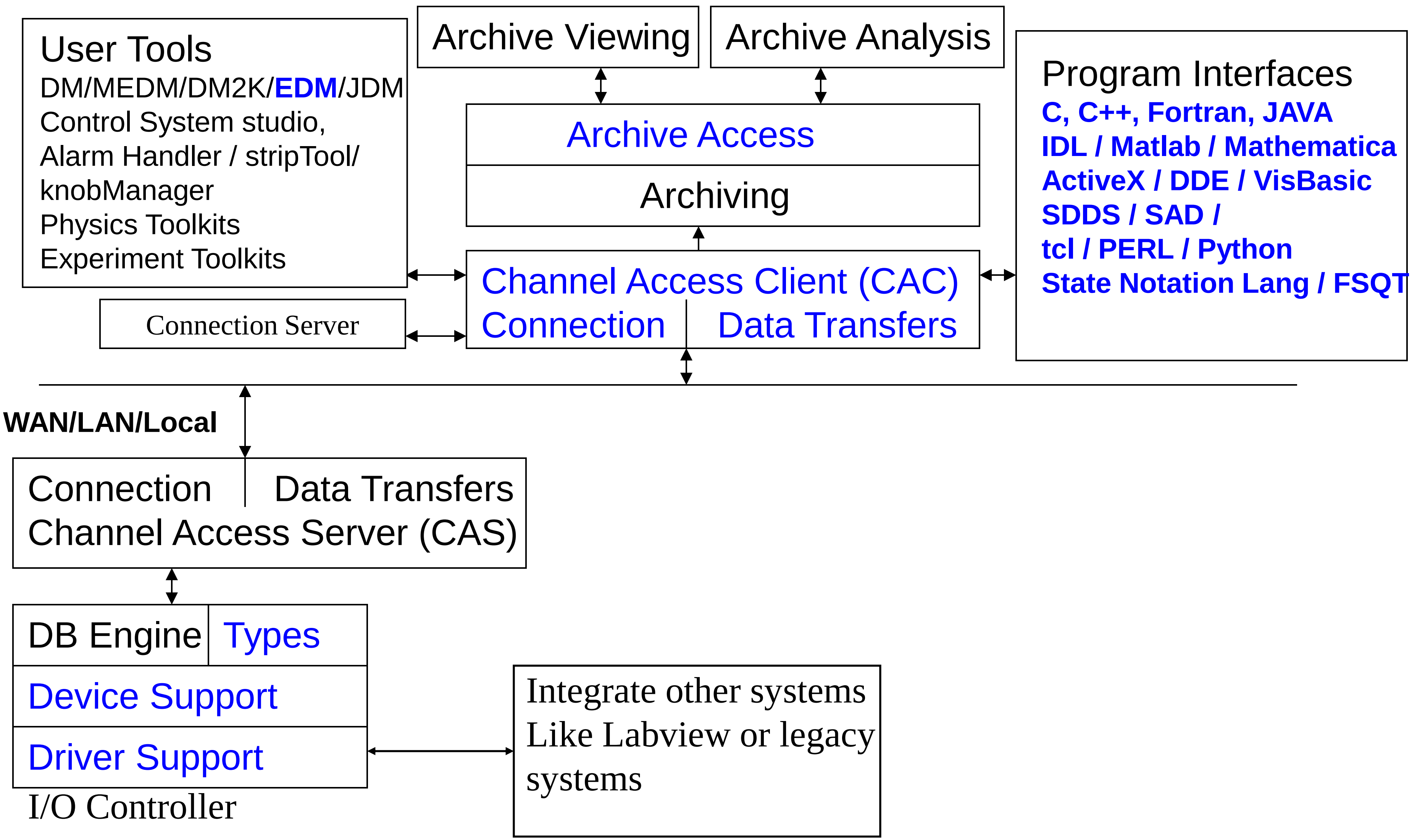
A data archiver with web based management and a suite of viewing and analysis capabilities

EPICS Architecture

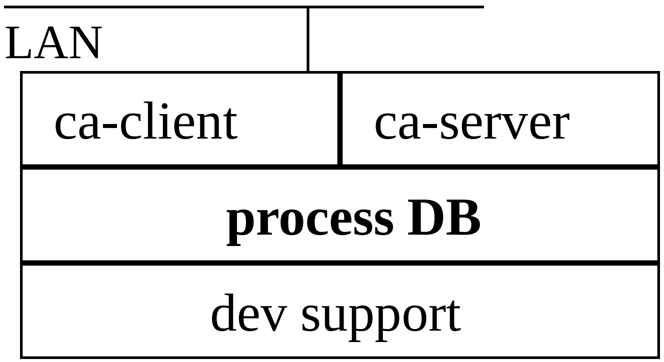


Programming interfaces for creating new channel access clients through standard language interfaces, scripting languages, specialized languages for state transition implementations, mathematical and modeling languages, and specialized accelerator analysis and control platforms

EPICS Architecture



IOC Core: Process Database



A Channel Access server provides connection, get, put, and monitor services to this database

A Channel Access client provides access to process DBs in other IOCs

Process Blocks are the basic elements: AI, AO, BI, BO, Motor, CALC, PID, SUB, etc.…

Process Blocks consist of fields for: SCHEDULE, I/O, CONVERT, ALARM, MONITOR

They hold runtime values: VALUE, TIMESTAMP, ALARM CONDITION, etc.…

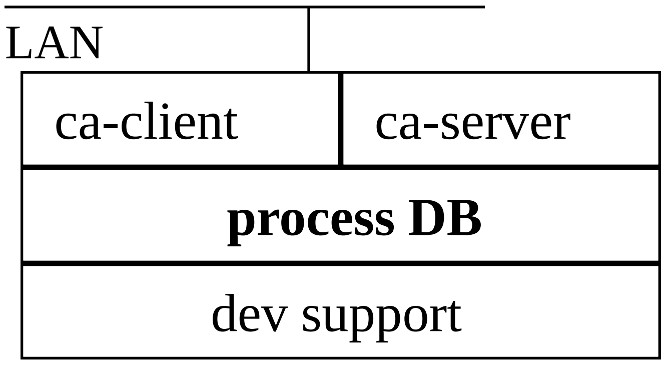
New process block are easily added

Configured using VDCT, GDCT, Relational DB, Text Editor at the workstation

Loaded as ASCII records at boot time

All fields can be read/written through the channel access client interface during operation

IOC Core: Process Database



Process Block execution time varies from block type to block type

AI on a PowerPC ~100,000/second (50% idle)
AI on a 68060 is ~18,000/second (50% idle)

Fastest periodic scan rate is dependent on OS clock tick – 1 KHz

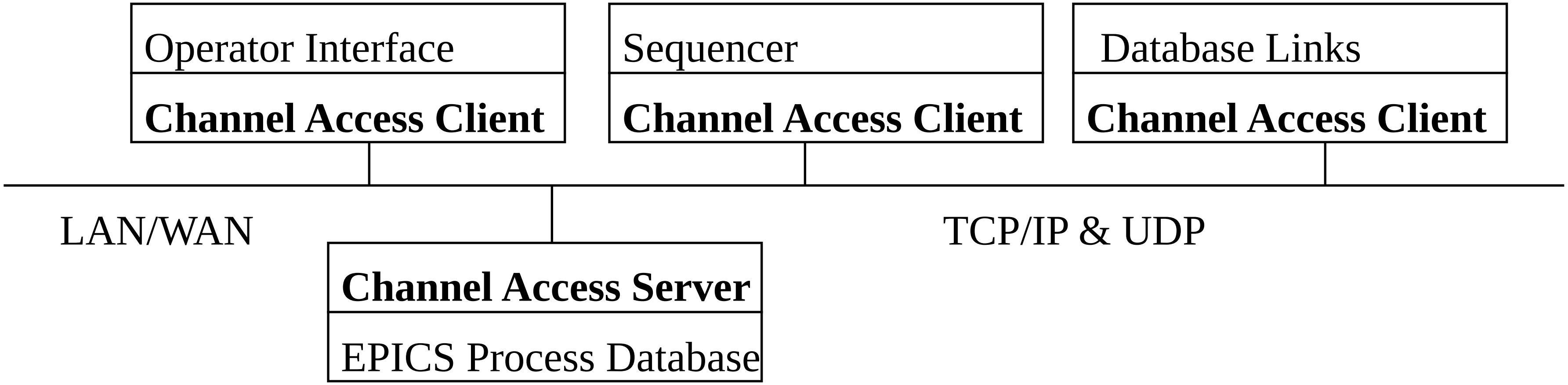
Interrupt scanning is limited by the CPU bandwidth (interrupt delay ~15usec)

Name resolution - 100,000/second - runs at the lowest priority

Support to particular physical I/O is distinct from process logic - ASCII device type

Platforms: UNIX, vxWorks, VMS (Client only), Windows NT, RTEMS, rtLinux

IOC Core: Channel Access Client/Server Libraries



Client: Provides read/write connections to any subsystem on the network with a channel access server

Server: Provides read/write connections to information in this node to any client on the network through channel access client calls. The data resides here!

Services: Dynamic Channel Location, Get, Put, Monitor
Access Control, Connection Monitoring, Automatic Reconnect
Conversion to client types, Composite Data Structures

Platforms: UNIX, vxWorks, VMS (Client only), Windows NT, RTEMS, rtLinux

IOC Core: Channel Access Services

Operator Interface
Channel Access Client

Sequencer
Channel Access Client

Database Links
Channel Access Client

TCP/IP & UDP
LAN/WAN

Channel Access Server
EPICS Process Database

Performance:
Over 1 Gbit Ethernet
Gets

	Propagation Delay	35 usec
	Throughput	780K /sec

Puts

	Propagation Delay	50 usec
	Throughput	700K /sec

Monitors

	Propagation Delay	Dependent
	Throughput	1.5M / sec

(Typically 10% channels have monitors)
(3% network load for 150K monitors / second)

Increase bandwidth with Routers, Bridges, Higher speed networks and EPICS gateway

Database: Minimum Configuration - Analog Input

ValveRB	
SCAN	*.1 Second*
INP	*#C0 S0*
DTYP	*XY566*
LINR	*Linear*
EGUF	*100*
EGUL	*0*
EGU	*%*

Analog Input

Reads the value for logical card 0 logical signal 0 Every .1 second from the Xycom566 device and scales the raw input between 0 and 100 percent. The raw value read from the hardware is put into the RVAL field and the converted value is put into the VAL field.

NOTE: device support sets the engineering units slope given EGUF and EGUL. The device support layer knows the number of bits of resolution of the hardware and the format

Database: Minimum Configuration - Discrete Input

TwoBitValveRB	
SCAN	*I/O Intr*
INP	*#C0 S4*
DTYP	*UniDig24*
NOBT	*2*
ZRVL	*0*
ONVL	*1*
TWVL	*2*
THVL	*3*
ZRST	*Travelling*
ONST	*Open*
TWST	*Closed*
THST	*Error*
THSV	*Major*

Mulibit Binary Input

Reads the value for logical card 0 bits 4 and 5 (counting from 0) every time there is any change on the UniDig24 card. The result to the four possible states. Both limits closed is an error condition.

IOC Core: Mapping Records to Channels

Record:
AI:
Name
SCAN
VAL

STAT	ACK
SEVR	ACKT
HOPR	EGU
LOPR	
HIHI	HHSV
HIGH	HSV
LOW	LSV
LOLO	LLSV

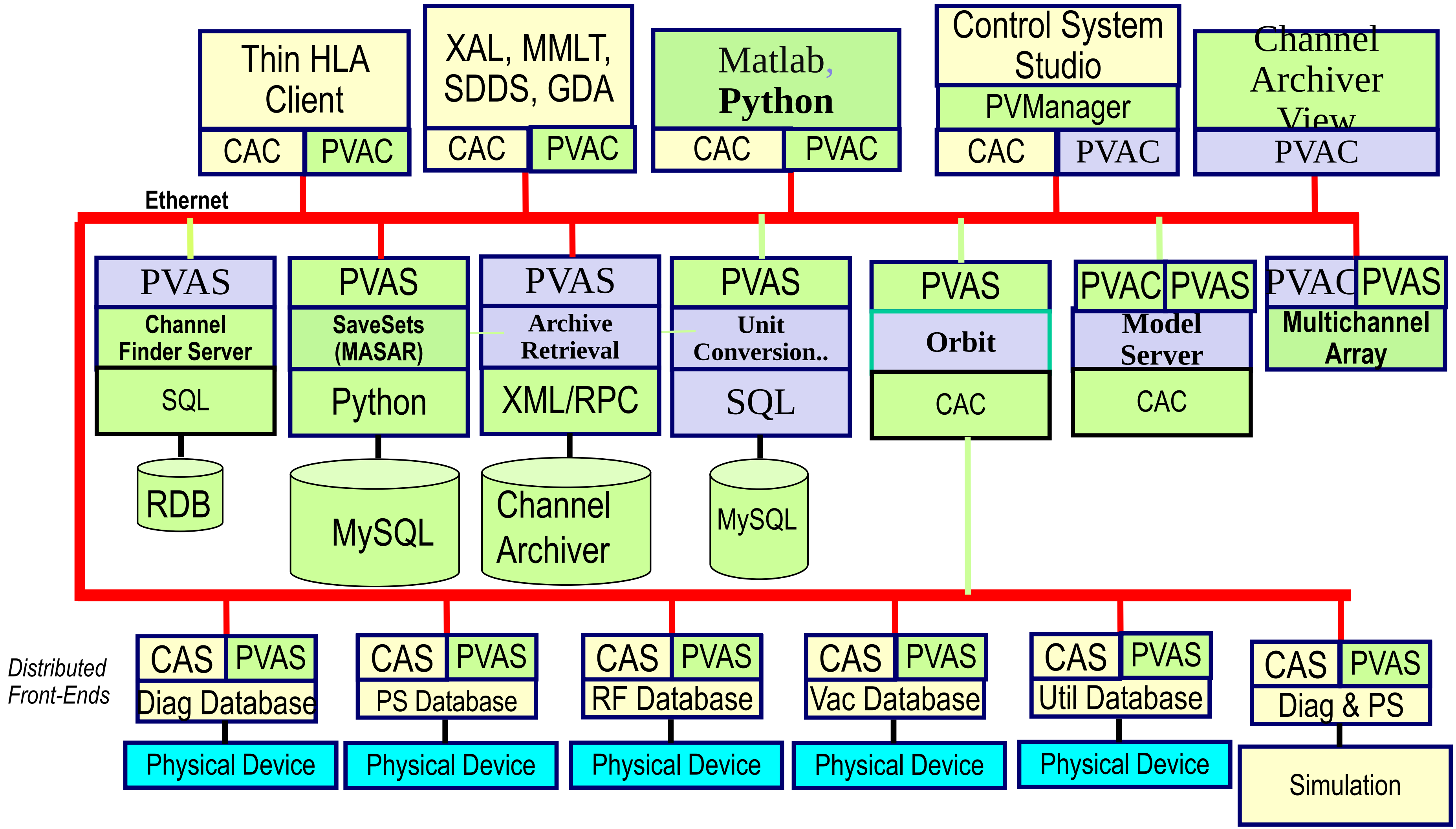
Channel Access Client:
Monitor request--------
Add Event to
- alarm change
- monitor change
- archive change

Make data type request
- Value
- Status & Severity
- Time Stamp
- Display and Control Information

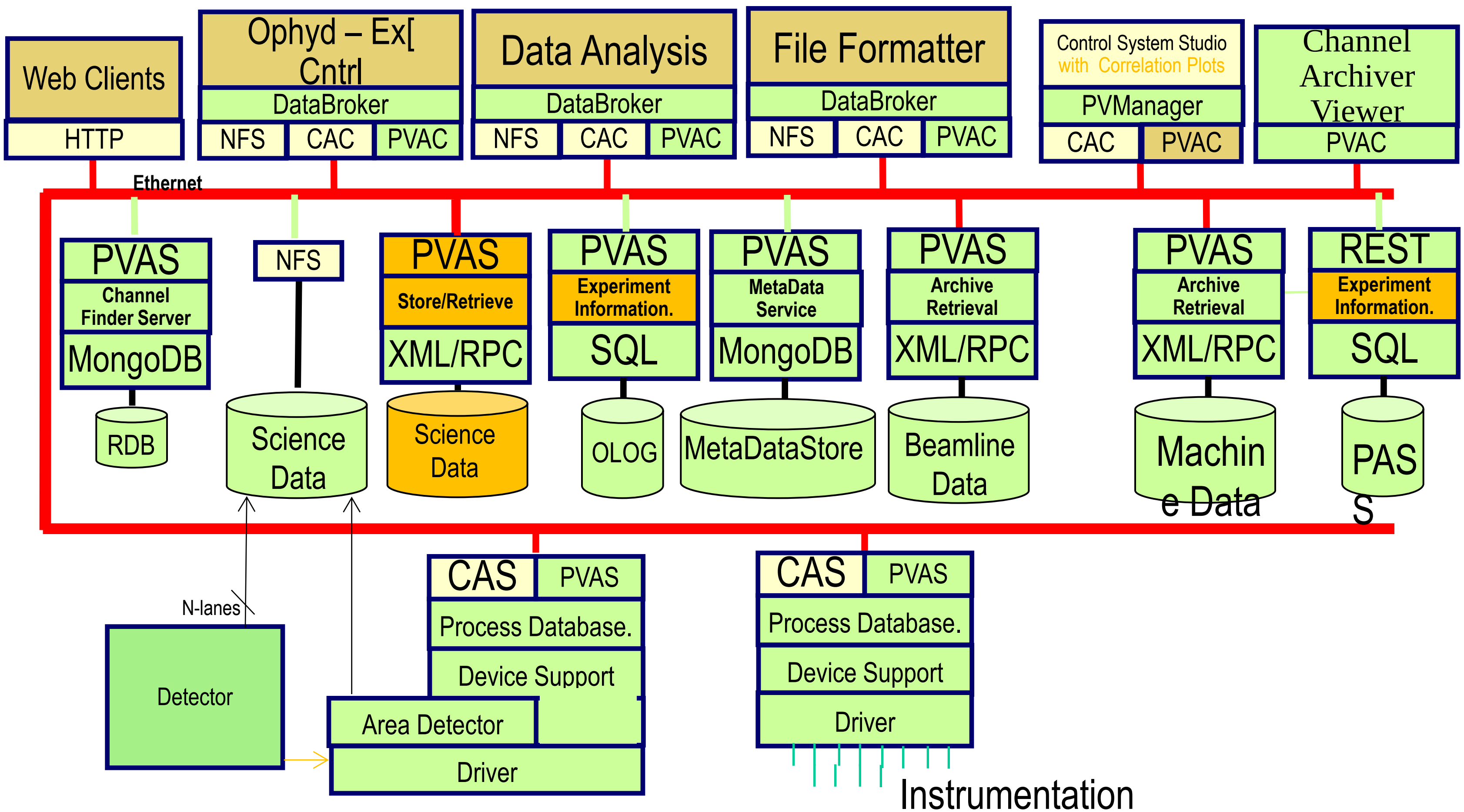
Channel Access Client:
Connect to
"LSX16a:M1:LVDT<.VAL>"
- LSX16a:M1:LVDT.VAL
- LSX16a:M1:LVDT.STAT
- LSX16a:M1:LVDT.SEVR
- LSX16a:M1:LVDT.TS
- LSX16a:M1:LVDT.HOPR
- LSX16a:M1:LVDT.LOPR
- LSX16a:M1:LVDT.EGU

Connect to
"LSX16a:M1:LVDT.SCAN"
- LSX16a:M1:LVDT.SCAN
- LSX16a:M1:LVDT.STAT
- LSX16a:M1:LVDT.SEVR
- LSX16a:M1:LVDT.TS
  - # Choices
  - List of choices

NSLS2 Physicists Use Services with Thin Apps



Experimental Data Architecture



The Learning Curve for EPICS

- Requires system administration skills
  - Installing EPICS –
  - Setting up the application environment to automatically build databases
  - Setting up the IOC to boot from the workstation
- Requires Programming Skills
  - Installing the new drivers
  - Knowledge of how to debug the application - is needed by everyone
  - Implementing state transition logic
- Some Instrumentation Knowledge
  - Learning to use the process database to implement data acquisition and steady state control such as PID control or interlocks.
- Must Have an Opinion
  - Choosing and learning which client tools to use

New Developments Take Advantage of Changes in Technology and Improve Functionality, Ease of Use, and Performance While Reducing Cost

- Version 4 of EPICS that supports extended functionality in record support and network communication which has hampered the ability to significantly change the communication protocol and maintain a large record type base. (SBIR grant)
- Continue to Develop new database configuration tools (APS/Cosy Lab/Diamond/SNS)
- Develop embedded EPICS support (NI, ZTEch, ITech, Yokogawa)
- Support for IOC core and hardware drivers under new operating systems (All labs)
- New High Level Physics Application Platform and online modeling (ORNL/LANL/CosyLab/BNL)

Conclusions: EPICS Continues to Meet the Needs of Its Members Through Cooperative Development of a Scaleable, Flexible Tool-kit

- The fundamental performance and functionality is scaleable and easily configured.
- Clean interfaces have been used to extend EPICS on the client side for varied operational tools, on the driver and device support side to integrate a wide variety of I/O and legacy systems.
- Open software development supports cooperative collaboration and gives members laboratories a larger pool of talent to support their controls requirements.
- Continual improvements allow members to expand functionality, performance, reliability and function while taking advantage of latest technology.